

WISE Receive Data

Description

The communication between two WISE devices is divided into two possibilities

- [Send / receive](#) a number
- [Send / receive](#) a data buffer

When number is send or received the data size is always 5 bytes - the first byte is a control byte (always 'L') and the remaining 4 bytes represents a 32 bit integer is a [little endian](#) format.

The data is sent with 512 byte segments. After each segment 4 byte response is received.

Maemo Platform

```
#define WISE_SEND_BUFFER_SIZE 512

int wise_recv_buffer(int socket,char** data,int* data_size)
{
    char tmp_buffer[WISE_SEND_BUFFER_SIZE];
    char buf[100];

    int transmitted=0,total_read=0;
    char* ptr=NULL;

    *data = NULL;
    *data_size=0;

    if ( data==NULL ) return WISE_CONNECTION_ERROR;

    /* receive data size */

    if ( wise_recv_int(socket,data_size)!=WISE_OK ) return WISE_CONNECTION_ERROR;

    /* check data size
       if ( *data_size <= 0 ) return WISE_CONNECTION_ERROR;

    /* allocate data buffer */
    *data = malloc(*data_size);
    if ( *data==NULL ) return WISE_CONNECTION_ERROR;
    memset(*data,0,*data_size);

    /* receive data */
    ptr = *data;
    while ( total_read < *data_size )
    {
        transmitted = read(socket,tmp_buffer,WISE_SEND_BUFFER_SIZE);
        if ( transmitted <= 0 ) return WISE_CONNECTION_ERROR;
        if ( total_read + transmitted > *data_size )
            return WISE_CONNECTION_ERROR;

        memcpy(ptr,tmp_buffer,transmitted);
        total_read += transmitted;
        ptr += transmitted;
    }
}
```

```
    strcpy(tmp_buffer,"ACK");
    transmitted = write(socket,tmp_buffer,4);
    if ( transmitted < 4 ) return WISE_CONNECTION_ERROR;
}
return WISE_OK;
}
```

S60 Platform

```
void CWISEBase::ReceiveBufferL(RSocket& aSocket, HBufC8** aBuffer)
{
TRequestStatus status;
 TInt totalRead(0), dataSize(0);
 TSockXfrLength len;

TBuf8<KWiseSendBufferSize> tmpBuffer;
TBuf8<4> ackBuffer;

ackBuffer.Copy(_L8("ACK"));
ackBuffer.AppendFill('\0',1);

// receive data size

dataSize = ReceiveIntL(aSocket);

// allocate data buffer
*aBuffer = HBufC8::NewL(dataSize);
TPtr8 bufferPtr(**aBuffer).Des();
bufferPtr.Fill('\0',dataSize);

// receive data
while ( totalRead < dataSize )
{
    TPtr8 ptr(bufferPtr.MidTPtr(totalRead));
    aSocket.RecvOneOrMore(tmpBuffer, 0, status, len);
    User::WaitForRequest(status);
    User::LeaveIfError(status.Int());

    ptr.Copy(tmpBuffer);
    totalRead += tmpBuffer.Length();

    aSocket.Send(ackBuffer, 0, status, len);
    User::WaitForRequest(status);
    User::LeaveIfError(status.Int());
}
}
```

Links

- [Archived:Wireless Information Sharing Engine](#)
- [WISE Protocol](#)

